

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | |
|--|---------------------------------|
| First Named Inventor : Stephen Appling | |
| Appln. No. : 09/747,366 | |
| Filed : December 22, 2000 | Group Art Unit: 2173 |
| Title : UPDATING OBJECTS CONTAINED WITHIN A WEBPAGE | Examiner: Dennis G. Bonshock |
| Docket No. : U75.12-0069/PA-057.11195-US | |

DECLARATION UNDER 37 C.F.R. §1.131

Mail Stop AF
Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SENT VIA EFS-WEB

I, Stephen C. Appling, hereby declare that:

1. I am the sole inventor of U.S. Patent Application Serial No. 09/747,366 ("the '366 application") referenced above.
2. The present application describes and claims methods for updating objects contained within a web page by providing an invisible frame on the web page that is configured to periodically request updated data from a server and cause an object in the web page to be updated based on the data acquired from the server.
3. I have reviewed Heidingsfeld et al., U.S. Patent No. 6,823,359, which was cited in the Office Actions mailed June 2, 2005, December 23, 2005, January 3, 2007, July 23, 2007, April 8, 2008 and January 21, 2009 in connection with the '366 application. I understand that Heidingsfeld et al. has an effective prior art date of November 21, 2000.
4. This declaration and the attached documents establish that I conceived and reduced to practice the system and method for updating objects contained within a webpage defined by pending claims 1-5, 7-11, 13 and 16-21 of the '366 application prior to November 21, 2000. The

documents are described below.

5. Attached as Exhibit 1 is a true and correct copy of an initial design document entitled "Greenhouse UI Architecture" in which the concept of updating objects contained within a web page using an invisible in-line frame is discussed. *See* Exhibit A, § 3.1. Exhibit 1 is an internal document that does not contain a date in the body of the document. However, the electronic file system on which it was created tracks both the date of creation, as well as the date the document was last modified. Exhibit 1 was last modified on January 25, 1999. Therefore, the content of Exhibit 1 that relates to the invention claimed in the '366 application was created at least as early as January 25, 1999.
6. Attached as Exhibit 2 is a true and correct copy of an advertisement from the January 2000 ASHRAE Journal announcing the release of a new product to be demonstrated at the International Air-Conditioning, Heating, Refrigerating Exposition ("AHR") Expo in Dallas, Texas on February 7 and 8 of 2000.
7. Attached as Exhibit 3 is a true and correct copy of an advertisement from the February 2000 ASHRAE Journal describing and showing the product, "WebCTRL", demonstrated at the AHR Expo on February 7 and 8 of 2000.
8. WebCTRL is a web based extension to building control system software, which provides an interactive graphical user interface to building operational and comfort conditions through a standard web browser. Built into WebCTRL is an embodiment of the invention claimed in the '366 application. Specifically, WebCTRL includes the functions to dynamically update objects in a web page using an invisible frame configured to periodically request information, such as sensor readings from a component of a building HVAC system, from a server and update the object with the information requested from the server.
9. A functioning prototype of WebCTRL, including the components for updating objects in a web page, was demonstrated at the AHR Expo on February 7 and 8 of 2000.
10. Although an embodiment of the invention claimed in the '366 application was included in the demonstration of WebCTRL at the AHR Expo, the invention was not publicly disclosed. The functions for dynamically updating objects contained within a web page were not publicly

disclosed at the AHR Expo, because it was only the results, i.e., objects on the page being updated, that were discernable through the public demonstration of the WebCTRL product. The functions or components of the WebCTRL product that cause the objects to update and that represent an embodiment of the invention claimed in the '366 application were not disclosed, but were present in the WebCTRL product and did operate as intended.

11. Attached as Exhibit 4 is a true and correct copy of a Java Server Page ("JSP") computer code file SSTAT.jsp last modified on November 24, 1999, with line numbers newly added for reference purposes.
12. Attached as Exhibit 5 is a true and correct copy of a JSP computer code file stdheader.jsp last modified on December 6, 1999, with line numbers newly added for reference purposes.
13. Attached as Exhibit 6 is a true and correct copy of a JSP computer code file stdfooter.jsp last modified on December 6, 1999, with line numbers newly added for reference purposes.
14. Attached as Exhibit 7 is a true and correct copy of a JSP computer code file StdActionPage.jsp last modified on December 6, 1999, with line numbers newly added for reference purposes.
15. Attached as Exhibit 8 is a true and correct copy of a JavaScript ("JS") computer code file Primitive.js last modified December 14, 1999, with line numbers newly added for reference purposes.
16. Attached as Exhibit 9 is a true and correct copy of a JS computer code file WidgetTextInput.js last modified on December 1, 1999, with line numbers newly added for reference purposes.
17. Attached as Exhibit 10 is a true and correct copy of a JS computer code file ControlTextInput.js last modified on November 29, 1999, with line numbers newly added for reference purposes.
18. Attached as Exhibit 11 is a true and correct copy of a Java computer code file PrimitiveUpdateServlet.java last modified on December 9, 1999, with line numbers newly added for reference purposes.
19. Attached as Exhibit 12 is a true and correct copy of a Java computer code file

ServletBase.java last modified on December 9, 1999, with line numbers newly added for reference purposes.

20. Attached as Exhibit 13 is a true and correct copy of a Java computer code file JscriptMsgList.java last modified on August 11, 1999, with line numbers newly added for reference purposes.
21. Attached as Exhibit 14 is a true and correct copy of a Java computer code file JscriptSimple.java last modified on December 2, 1999, with line numbers newly added for reference purposes.
22. Attached as Exhibit 15 is a true and correct copy of a Java computer code file JscriptSetMasterValue.java last modified on August 26, 1999, with line numbers newly added for reference purposes.
23. Attached as Exhibit 16 is a true and correct copy of a screen shot showing a last modified date of "SSTAT.jsp" (Exhibit 4) in a version control code database.
24. Attached as Exhibit 17 is a true and correct copy of a screen shot showing last modified dates of, *inter alia*, "stdheader.jsp" (Exhibit 5) and "stdfooter.jsp" (Exhibit 6) in a version control code database.
25. Attached as Exhibit 18 is a true and correct copy of a screen shot showing a last modified date of, *inter alia*, "StdActionPage.jsp" (Exhibit 7) in a version control code database.
26. Attached as Exhibit 19 is a true and correct copy of a screen shot showing last modified dates of, *inter alia*, "Primitive.js" (Exhibit 8), "WidgetTextInput.js" (Exhibit 9) and "ControlTextInput.js" (Exhibit 10) in a version control code database.
27. Attached as Exhibit 20 is a true and correct copy of a screen shot showing last modified dates of, *inter alia*, "PrimitiveUpdateServlet.java" (Exhibit 11) and "ServletBase.java" (Exhibit 12) in a version control code database.
28. Attached as Exhibit 21 is a true and correct copy of a screen shot showing last modified dates of, *inter alia*, "JscriptMsgList.java" (Exhibit 13), "JscriptSimple.java" (Exhibit 14) and "JscriptSetMasterValue.java" (Exhibit 15) in a version control code database.
29. In order to further evidence reduction to practice of the invention as claimed in the '366

application, a small subset of the source code used in the WebCTRL product, which was implemented using the programming languages Java and JavaScript, has been attached as Exhibits 4-15. The code exhibits have been modified solely to add line numbers every five lines in order to facilitate cross-referencing particular portions in the present document. Exhibits 16-21 are screen shots that show last modified dates for the code exhibits, thereby confirming that the appended code, evidencing reduction to practice of the present invention, was in existence no later than February 7, 2000.

30. The following paragraphs contain a general discussion of how the appended code of the WebCTRL product operated in relation to the invention claimed in the '366 application, which was designed to work with a HVAC system.
31. Part of the WebCTRL system generated web pages as Java Server Pages (JSPs) to provide access to custom programming in building automation field control modules. These generated pages all had a similar structure. The file "SSTAT.jsp" (Exhibit 4) was an example of one of these types of files. It began with an include of stdheader.jsp (Exhibit 5) and ended with an include of stdfooter.jsp (Exhibit 6). (*See* Exhibit 4 lines 1 and 57). The bulk of the file consisted of includes for each type of "microblock" used in the custom programming of the field module. A "microblock" represented a standardized programming element in a field module such as a physical analog input or a Proportional/Integral/Derivative ("PID") control loop. These microblock includes added JavaScript controls to display or edit dynamic values in the system.
32. Stdheader.jsp (Exhibit 5) had several important components. Lines 12-42 of Stdheader.jsp included javascript source files for dynamically updateable controls. Line 44 of Stdheader.jsp included StdActionPage.jsp (Exhibit 7) on the web server. Lastly, Stdheader.jsp included the three IFRAME elements used to communicate with the server without reloading the main web page. Notably, these IFRAMEs had both a height and width of "0", which made them invisible in the web browser on that page.
33. Dynamically updated values were represented in JavaScript using "Primitive" objects defined in Primitive.js (Exhibit 8). On screen displays of these values was accomplished in two parts: the actual rendering of a value to HTML was handled in a "widget" (such as the text input widget

defined in WidgetTextInput.js (Exhibit 9)) and the control (such as the number edit control defined in ControlTextInput.js (Exhibit 10)) watched for changes to a primitive and sets the value in the widget.

34. The primitiveupdate IFRAME (stdheader.jsp (Exhibit 5) line 53) was refreshed automatically (by a timer) and used to poll the server for updated values. This was configured in StdActionPage.jsp (Exhibit 7) lines 37-53. The getUpdate function caused the primitiveupdate IFRAME to reload the address “/common/servlet/primitiveupdate”. This returned JavaScript which was executed in the IFRAME. The last line of this JavaScript would have been a call to the “updateFinished” function (StdActionPage.jsp lines 40-44) which set a timer to call getUpdate again.
35. The address “/common/servlet/primitiveupdate” that was constantly polled by the primitiveupdate IFRAME was mapped to a Java Servlet on the web server, the PrimitiveUpdateServlet (defined in PrimitiveUpdateServlet.java (Exhibit 11)). This polling resulted in a call to the “doGet” method (PrimitiveUpdateServlet.java line 70). The “doGet” method: established a valid session (PrimitiveUpdateServlet.java line 84), got changed values into a list of JavaScript messages (PrimitiveUpdateServlet.java line 110), and formatted this list to return to the web browser (PrimitiveUpdateServlet.java line 128). Note the addition of the call to “updateFinished” which kept the IFRAME continually refreshing on line 121 of PrimitiveUpdateServlet.java.
36. The call to “stdContent” on line 128 of PrimitiveUpdateServlet.java resulted in a call to JScriptMsgList.write (ServletBase.java (Exhibit 12) line 417). This “write” method (JScriptMsgList.java (Exhibit 13) lines 107-121) iterated over each JscriptMsg element in its list and called “encodeHtml” on each one to write the resulting JavaScript back to the IFRAME in the browser.
37. Normal value changes for a simple control were represented in the JavaScript message list as a JscriptSetMasterValue which was derived from JscriptSimple (Exhibit 14). The “encodeHTML” method (JscriptSimple.java line 95) converted the JscriptSetMasterValue into a string and sent it back to the browser. JscriptSetMasterValue generated the JavaScript

“window.parent.PrimitiveBase.setPrimitiveMasterValue” with parameters for the ID of the primitive and the updated value (JScriptSetMasterValue.java (Exhibit 15) lines 45-49). This was executed in the primitiveupdate IFRAME but it referred to the main HTML page using “window.parent” and set the value of the JavaScript primitive object there.


38. The function “setPrimitiveMasterValue” called by the IFRAME was defined in Primitive.js (See Exhibit 8, lines 661-665 and 761). It looked up the appropriate primitive object and called “setMasterValue(newData)” on it (Primitive.js line 664). “setMasterValue” (Primitive.js lines 325-339) checked if the value had changed and then called “notifyListeners” (Primitive.js line 327). “notifyListeners” (Primitive.js lines 280-289) iterated over a list of listeners and called “valueChanged” on each one. One of these listeners was the JavaScript control object associated with this value (such as the ControlNumberEdit object defined in ControlTextInput.js (Exhibit 10) lines 595-894).
39. The ControlNumberEdit object inherited some methods from a prototype ControlTextInput object (ControlTextInput.js line 705) including the “valueChanged” method (ControlTextInput.js lines 372-421 and 494). The “valueChanged” function found the associated widget object (in this case a WidgetTextInput) and called “setValue(value)” on it (ControlTextInput.js line 408). WidgetTextInput.setValue assigned the value to the actual HTML element that represented the value on the page on WidgetTextInput.js line 603. This entire chain caused a server side value change to be displayed on the main web page without refreshing it. Only the hidden IFRAME was updated.
40. As evidenced by the design document of Exhibit 1, the advertisements of Exhibits 2 and 3, and the code documents of Exhibits 4-15 (plus the date verification of the code documents provided by Exhibits 16-21), the invention claimed in the ‘366 application was conceived and reduced to practice at least as early as February 7, 2000, which is prior to the November 21, 2000 prior art date of Heidingsfeld et al.
41. The documents provided as Exhibits 4-21 were confidential, and not available to the public. The actually code embodied in the WebCTRL product was not made public at the AHR Expo in February of 2000.

First Named Inventor: Stephen Appling

Application No.: 09/747,366

-8-

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statement were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.



Stephen C. Appling

Date: 5/11/2009